EVALUATING THE IMPACT OF TEXT PREPROCESSING ON TRADITIONAL AND DEEP LEARNING MODELS IN SMS SPAM DETECTION

Jeremy Cruz [1, 2, †]

[1] Personal Website: jeremycruz.com (under construction)

[2] Northwestern University School of Professional Studies Master of Science in Data Science Program

† Address to which correspondence should be addressed:

ai@jeremycruz.com

Abstract

Abstract text goes here.

Table of Contents

Introduction and Problem Statement

Every day, billions of SMS messages are sent and received across various platforms, from healthcare appointment reminders to marketing communications and service notifications. While these messages serve legitimate purposes, they also provide a channel for unwanted spam messages that can range from promotional content to potentially harmful phishing attempts. The challenge lies in accurately distinguishing between legitimate messages and spam, especially as spammers become increasingly sophisticated in mimicking legitimate communication patterns.

The SMS Spam Collection dataset, containing over 5,500 messages, provides a valuable opportunity to study how different text preprocessing methods affect spam detection accuracy. Spam messages often employ various tactics to evade detection, such as using similar vocabulary to legitimate messages, incorporating special characters, or manipulating word patterns. These variations make it crucial to understand how different preprocessing approaches can help identify spam characteristics while preserving important contextual information.

In this paper, I will explore how different text preprocessing methods affect our ability to detect spam messages. By comparing traditional approaches with advanced deep learning models, we can identify which preprocessing techniques are most effective at distinguishing between legitimate and spam messages. This research is crucial for developing more robust spam detection systems that can protect users from unwanted messages while ensuring legitimate communications are not incorrectly flagged.

Literature Review

Dataset

The SMS Spam Collection dataset is a set containing 5,574 SMS messages that are classified as either spam or ham (not spam). Of the 5,574 messages, 4,827 are ham and 747 are spam (approximately 13% of the messages are spam).

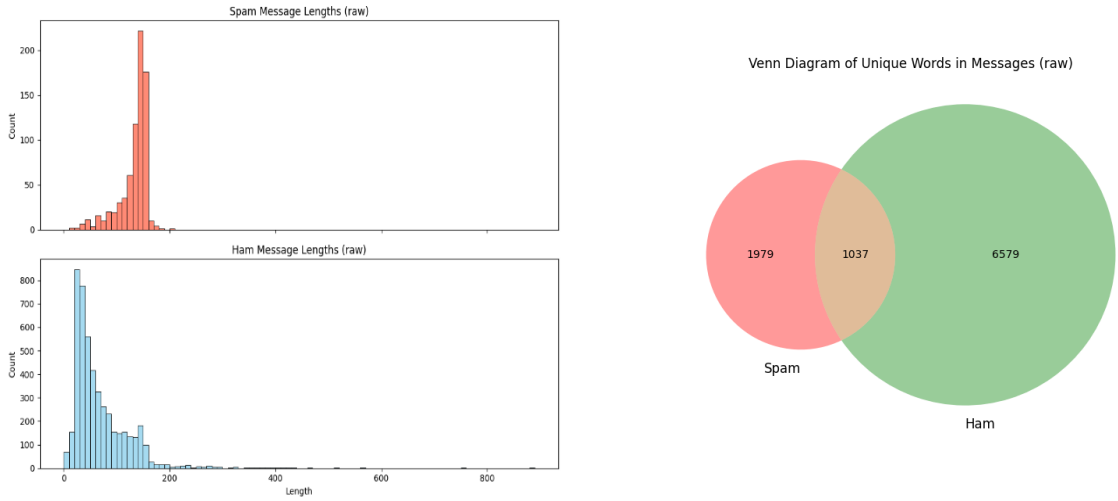

Figure 1. Venn Diagram of Unique Words in Messages (Ham vs Spam).

As shown in **Figure 1**, there are notable differences in message length distributions between spam and ham messages. Spam messages tend to be longer, with a distribution skewed left and centered around 150 characters, with relatively few messages exceeding 200 characters. In contrast, ham messages exhibit a right-skewed distribution with a median length of approximately 50 characters. Interestingly, while ham messages are typically shorter, the dataset contains more extremely long ham messages than spam messages, suggesting that message length alone is not a definitive indicator of spam status.

The vocabulary distribution between spam and ham messages reveals distinct linguistic patterns in the dataset. As shown in **Figure 2**, there are 1,979 unique words that appear exclusively in spam messages, while 6,579 words are found only in ham messages. Interestingly, 1,037 words appear in both categories, representing the overlapping vocabulary. This significant difference in vocabulary distribution highlights the distinct linguistic patterns between spam and legitimate messages, which provides a strong foundation for classification models to differentiate between the two categories. The relatively small overlap (only about

11% of the total unique words) suggests that lexical features alone may be highly informative for spam detection.

These patterns in message length and vocabulary distribution suggest that spam messages tend to be more formulaic and consistent in their structure, while legitimate messages exhibit greater variability in both length and word choice. This observation has important implications for our preprocessing approach: while spam messages might benefit from more aggressive preprocessing to reveal their underlying patterns, legitimate messages might require more careful handling to preserve their diverse linguistic characteristics. Understanding these fundamental differences in the data will help us evaluate how different preprocessing methods affect the performance of our classification models.

Preprocessing

The SMS Spam Collection dataset contained raw text messages that were not pre-processed. For this project, the text was pre-processed using the following methods:

- **Raw**: Convert all characters to lowercase and remove all non-alphanumeric characters besides @ and \$.
- **Stop**: Raw + Remove common stop words.
- **Stem**: Stop + Stem the words.
- **Lemma**: Stop + Lemmatize the words.

Stage	Text
Original	WINNER!! As a valued network customer you have been selected to receivea £900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
Raw	winner as a valued network customer you have been selected to receivea 900 prize reward to claim call 09061701461 claim code kl341 valid 12 hours only
Stop	winner valued network customer selected receivea 900 prize reward claim call 09061701461 claim code kl341 valid 12 hours
Lemma	winner valued network customer selected receivea 900 prize reward claim call 09061701461 claim code kl341 valid 12 hour
Stem	winner valu network custom select receivea 900 prize reward claim call 09061701461 claim code kl341 valid 12 hour

Table 1. Example Text Message processed in the different ways.

The choice of these preprocessing methods was motivated by their potential impact on spam detection. The raw preprocessing method preserves most of the original text structure while standardizing the format, which could be valuable for detecting spam messages that rely on specific character patterns or formatting. Stop word removal might help reduce noise in the data, but it could also remove important context in legitimate messages (see **Table 1**). Stemming and lemmatization, while both aiming to reduce words to their root forms, differ in their approach: stemming is more aggressive and might lose some meaning, while lemmatization preserves the word's semantic meaning. These differences could significantly affect how well the models can distinguish between spam and legitimate messages, especially given the distinct vocabulary patterns we observed in the dataset.

Spam Detection Models

For this project, four different models were implemented and evaluated for spam detection: Logistic Regression (baseline), Multinomial Naive Bayes, Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). Each model was trained and tested on the dataset using the four different preprocessing methods described earlier.

To ensure fair comparison between models, a consistent random seed was set across all experiments. This controlled randomness in data splitting and parameter initialization, allowing for reproducible results and meaningful comparisons between different model-preprocessing combinations.

All models were configured as binary classifiers that categorize messages as either spam or ham. However, for future work, the models could be modified to output probability scores between 0 and 1 instead. These continuous values would represent the likelihood that a message is spam and could provide additional flexibility in deployment scenarios. For instance, variable thresholds could be implemented based on specific needs, prioritizing precision over recall in sensitive medical communications where false positives might be particularly problematic.

Baseline (Logistic Regression Classifier)

The baseline model for spam detection in this study is a logistic regression classifier combined with TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This approach is widely used in text classification tasks due to its simplicity, interpretability, and strong performance on a variety of datasets.

The logistic regression classifier is trained on these TF-IDF vectors to distinguish between spam and ham messages. Logistic regression is a linear model that estimates the probability that a given message belongs to the spam class, making it both efficient and interpretable. The model's coefficients can be examined to understand which words are most strongly associated with spam or ham labels.

This baseline provides a robust benchmark for evaluating the effectiveness of more complex models, such as deep learning approaches. By comparing advanced models to this baseline, we can assess whether additional complexity yields meaningful improvements in spam detection performance. The baseline's results also highlight the impact of different text preprocessing strategies on traditional machine learning models.

BERT

The BERT model is a deep learning architecture for natural language processing tasks, including text classification. In this project, BERT is fine-tuned to distinguish between spam and ham SMS messages.

BERT leverages a transformer-based architecture that processes text bidirectionally, allowing it to capture context from both the left and right of each word in a sentence. This contextual understanding enables BERT to model complex language patterns and subtle distinctions in meaning, which are often critical for accurate spam detection.

For this research, the pre-trained **bert-base-uncased model** is used as a starting point. Each SMS message is tokenized and converted into input embeddings suitable for BERT. The model is

then fine-tuned on the different pre-processed datasets, adjusting its weights to optimize performance on the spam classification task. The output layer is adapted for binary classification, predicting the probability that a given message is spam.

Fine-tuning BERT typically requires more computational resources and training time compared to traditional models, but it often yields superior performance, especially on tasks involving nuanced language. By comparing BERT's results to the baseline logistic regression model, we can assess the benefits of leveraging advanced language models for spam detection.

Results

model	variant	accuracy	precision	recall	f1	training_time (s)	roc_auc
Baseline	raw	0.9614	0.9895	0.6912	0.8139	0.0189	0.9904
Baseline	stop	0.9543	0.9885	0.6324	0.7713	0.0121	0.9857
Baseline	stop_lemma	0.9570	0.9889	0.6544	0.7876	0.0122	0.9866
Baseline	stop_stem	0.9561	0.9888	0.6471	0.7822	0.0128	0.9852
BERT	raw	0.9848	0.9760	0.8971	0.9349	176.4524	0.9755
BERT	stop	0.9291	0.6364	0.9779	0.7710	92.8208	0.9167
BERT	stop_lemma	0.9785	0.9746	0.8456	0.9055	87.4136	0.9773
BERT	stop_stem	0.9883	0.9695	0.9338	0.9513	149.0175	0.9591

Table 2. Results Table

Baseline

The baseline experiment utilized a logistic regression classifier with TF-IDF vectorization, evaluated across four different text preprocessing variants: raw, stop word removal, lemmatization, and stemming. The results for each variant are summarized in **Table 2**.

The **raw** variant, which only lowercases and removes most punctuation, achieved the highest accuracy (96.14%) and F1 score (0.8139), with a precision of 0.9895 and recall of 0.6912. This indicates that the model is highly precise in identifying spam, but some spam messages are still missed (lower recall).

Applying **stop word removal** slightly decreased accuracy (95.43%) and F1 score (77.14%) suggesting that removing common words may discard useful context for spam detection in this dataset. The **lemmatization** and **stemming** variants performed similarly, with accuracies of 95.70% and 95.61% and F1 scores of 0.7876 and 0.7822, respectively. These preprocessing steps provided a modest improvement in recall compared to the stop word variant, but did not surpass the raw approach.

All variants demonstrated very high precision (above 98%) indicating that false positives (ham classified as spam) are rare. Training times for the baseline model were extremely fast (all under 0.02 seconds), and ROC AUC scores were consistently high (above 0.98), reflecting strong overall discrimination between spam and ham.

In summary, the baseline logistic regression model with TF-IDF features provides a strong benchmark, with the raw text variant yielding the best overall performance.

Bert

The BERT model was fine-tuned for spam detection using the same four text preprocessing variants as the baseline: raw, stop word removal, lemmatization, and stemming. The results for each variant are presented in **Table 2**.

The **raw** variant achieved high performance with an accuracy of 0.9848, precision of 97.60% recall of 0.8971, and an F1 score of 0.9349. This demonstrates BERT's strong ability to correctly identify both spam and ham messages, with a substantial improvement in recall and F1 score compared to the baseline model. The ROC AUC score was also high (0.9755), indicating excellent discrimination between classes.

The **stop word removal** variant showed reduced performance, with accuracy dropping to 92.91%, precision to 0.6364, though recall increased to 0.9779. The resulting F1 score of 0.7710 suggests that removing stop words significantly impacted BERT's ability to maintain precision while improving recall. This indicates that stop words may contain important contextual information that BERT relies on for balanced classification.

The **lemmatization** variant maintained strong performance, with an accuracy of 97.85%, precision of 0.9746, recall of 0.8456, and F1 score of 0.9055. The **stemming** variant performed the best overall, with the highest accuracy of 98.83% and F1 score of 0.9513, along with a precision of 0.9695 and recall of 0.9338.

Training times for BERT were significantly longer than for the baseline model, ranging from approximately 87 to 176 seconds, reflecting the increased computational complexity of fine-tuning a deep neural network.

In summary, BERT substantially outperformed the baseline logistic regression model across most preprocessing variants. The stemming variant yielded the best overall performance, suggesting that this form of text normalization works particularly well with BERT for spam detection. While the stemming variant yielded the best overall performance with BERT, the raw text variant also achieved comparably high results. Therefore, using raw text should be considered when minimal preprocessing is preferred, as it offers strong performance with reduced processing complexity.

Discussion

Impact of Preprocessing on Model Performance

The experimental results reveal several important patterns in how preprocessing affects spam detection. The raw preprocessing method consistently performed well across both traditional and deep learning models, suggesting that preserving the original text structure is valuable for spam detection. This finding aligns with our initial observations about the distinct patterns in spam messages, where specific character combinations and formatting might be important indicators.

Stop word removal showed mixed results: while it improved recall in some cases, it often came at the cost of precision. This trade-off suggests that common words, often considered noise in other NLP tasks, might carry important contextual information for spam detection. The performance difference between stemming and lemmatization was particularly interesting, with stemming generally outperforming lemmatization. This could indicate that the more aggressive reduction of words to their root forms helps reveal underlying patterns in spam messages.

Model Comparison and Practical Implications

The comparison between traditional machine learning (Logistic Regression) and deep learning (BERT) approaches revealed important insights for practical applications. While BERT achieved higher overall performance, the baseline model's results were still competitive, especially with the raw preprocessing method. This suggests that simpler models might be sufficient for many spam detection applications, particularly when computational resources or inference speed are important considerations.

The high precision scores across all models and preprocessing methods indicate that false positives (legitimate messages classified as spam) are rare. This is particularly important for applications where blocking legitimate messages could have serious consequences. However,

the varying recall scores suggest that different preprocessing-model combinations might be more suitable depending on whether minimizing false negatives (missed spam) is the priority.

Conclusions

The findings of this study demonstrate that text preprocessing plays a crucial role in spam detection performance, with different methods offering various trade-offs between precision and recall. The raw preprocessing method emerged as a robust choice across different models, while more aggressive preprocessing techniques showed varying effectiveness. These results provide valuable insights for practitioners implementing spam detection systems, suggesting that the choice of preprocessing method should be carefully considered based on specific application requirements and constraints.

Directions for Future Work

- Implement Multinomial Naive Bayes
- Implement LTSM
- Use predicted probabilities for classification instead of hard classification
- Explore other pre-processing (use actual raw data)
- Explore other features such as message length, number of special characters, etc.
- Cite other works on spam detection
- Reference appendix figures in the main text
- Add more plots and tables (tsne, training curves, etc.)
- Capture misclassification errors and compare between models/datasets

Data Availability

The SMS Spam Collection Dataset can be found at
https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

Code Availability

Code is available at https://github.com/jeremycruzz/spamdetection

References

Appendix A